The Loudness of Spotify Tracks Over Time

Data Setup | **Visualization** | **Model and Conclusion**

Data Setup

[1] Selected Dataset

Spotify Dataset 1921-2020, 600k+ Tracks

[2] Project Description

Problem

In the modern streaming era, artists are fighting for listeners' limited attention spans. A common theory in the music industry, known as the "Loudness War," suggests that producers have been progressively mixing songs louder to make them stand out on radio and playlists. But is this trend anecdotal, or can it be proven with data? My project investigates the evolution of song loudness over the last century to determine if popular music is statistically getting louder or quieter.

Data

This project uses the 'Spotify Dataset 1921-2020, 600k+ Tracks' from Kaggle by Yamac Eren Ay (Updated in 2022). By isolating loudness and duration, these song attributes can be analyzed over the course of 100 years.

Methodology

1. Data Cleaning: Filtering out podcasts, unreasonably long songs, and missing release date information.
2. Exploratory Data Analysis (EDA): Visualizing trends using heatmaps and histograms to identify correlations.
3. Predictive Modeling: Training a simple linear regression model to predict and quantify the rate at which music is getting louder.

[3] Checking the Data

[3.1] Summary

In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

df = pd.read_csv('data/tracks.csv')

print('Shape is:', df.shape)
df.info(verbose=True)
df.describe()
```

```
Shape is: (586672, 20)
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 586672 entries, 0 to 586671
Data columns (total 20 columns):
 #   Column            Non-Null Count   Dtype  
---  ------            --------------   -----  
 0   id                586672 non-null  object 
 1   name              586601 non-null  object 
 2   popularity        586672 non-null  int64  
 3   duration_ms       586672 non-null  int64  
 4   explicit          586672 non-null  int64  
 5   artists           586672 non-null  object 
 6   id_artists        586672 non-null  object 
 7   release_date      586672 non-null  object 
 8   danceability      586672 non-null  float64
 9   energy            586672 non-null  float64
 10  key               586672 non-null  int64  
 11  loudness          586672 non-null  float64
 12  mode              586672 non-null  int64  
 13  speechiness       586672 non-null  float64
 14  acousticness      586672 non-null  float64
 15  instrumentalness  586672 non-null  float64
 16  liveness          586672 non-null  float64
 17  valence           586672 non-null  float64
 18  tempo             586672 non-null  float64
 19  time_signature    586672 non-null  int64  
dtypes: float64(9), int64(6), object(5)
memory usage: 89.5+ MB
```

Out[1]:

	popularity	duration_ms	explicit	danceability	energy	
count	586672.000000	5.866720e+05	586672.000000	586672.000000	586672.000000	586672.0
mean	27.570053	2.300512e+05	0.044086	0.563594	0.542036	5.2
std	18.370642	1.265261e+05	0.205286	0.166103	0.251923	3.5
min	0.000000	3.344000e+03	0.000000	0.000000	0.000000	0.0
25%	13.000000	1.750930e+05	0.000000	0.453000	0.343000	2.0
50%	27.000000	2.148930e+05	0.000000	0.577000	0.549000	5.0
75%	41.000000	2.638670e+05	0.000000	0.686000	0.748000	8.0
max	100.000000	5.621218e+06	1.000000	0.991000	1.000000	11.0

[3.2] Cleaning the Dataset

[3.2.1] Convert duration from ms (int) to minutes (float)

In [2]:
```
df['duration_min'] = df['duration_ms'] / 60000
```

[3.2.2] Convert date from string obj to year (int)

In [3]:
```
# Remove rows without release dates
df = df.dropna(subset=['release_date'])

# Convert ISO8601 date YYYY-MM-DD to just YYYY
df['release_year'] = pd.to_datetime(df['release_date'], format='ISO8601').dt.year
```

[3.3] Creating a Subset

In [4]:
```
# Attempt to trim off excessively long tracks like podcasts

df_sub = df[
    (df['duration_min'] < 15) &
    (df['release_year'] >= 1921)
    ].copy()

df_sub.info(verbose=True)
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 584576 entries, 0 to 586671
Data columns (total 22 columns):
 #   Column            Non-Null Count   Dtype  
---  ------            --------------   -----  
 0   id                584576 non-null  object 
 1   name              584505 non-null  object 
 2   popularity        584576 non-null  int64  
 3   duration_ms       584576 non-null  int64  
 4   explicit          584576 non-null  int64  
 5   artists           584576 non-null  object 
 6   id_artists        584576 non-null  object 
 7   release_date      584576 non-null  object 
 8   danceability      584576 non-null  float64
 9   energy            584576 non-null  float64
 10  key               584576 non-null  int64  
 11  loudness          584576 non-null  float64
 12  mode              584576 non-null  int64  
 13  speechiness       584576 non-null  float64
 14  acousticness      584576 non-null  float64
 15  instrumentalness  584576 non-null  float64
 16  liveness          584576 non-null  float64
 17  valence           584576 non-null  float64
 18  tempo             584576 non-null  float64
 19  time_signature    584576 non-null  int64  
 20  duration_min      584576 non-null  float64
 21  release_year      584576 non-null  int32  
dtypes: float64(10), int32(1), int64(6), object(5)
memory usage: 100.3+ MB
```